

Kubeisa S, Ugail H and Wilson M (2004): "Interactive Design Using Higher Order PDE's " *The Visual Computer*, 20 (10): 682 - 693.

Interactive Design Using Higher Order PDEs

Sebastian Kubiesa¹, Hassan Ugail² and Michael Wilson³

¹ Department of Applied Mathematics, University of Leeds, Leeds LS2 9JT, UK,

² Department of Electronic Imaging and Media Communications, School of Informatics, University of Bradford, Bradford BD7 1DP, UK

³ Department of Applied Mathematics, University of Leeds, Leeds LS2 9JT, UK

Received: date / Revised version: date

Key words Interactive Design, PDE surfaces, Higher Order PDEs

Abstract This paper extends the PDE method of surface generation. The governing partial differential equation is generalised to sixth order to increase its flexibility. The PDE is solved analytically, even in the case of general boundary conditions, making the method fast. The boundary conditions, which control the surface shape, are specified interactively, allowing intuitive manipulation of generic shapes. A compact user interface is presented which makes use of direct manipulation and other techniques for 3D interaction.

1 Introduction

A three-dimensional (3D) computer aided design (CAD) system should facilitate easy and accurate creation of free-form surfaces [29]. The system should also contain a description of the underlying geometric structure of the design to allow alternative configurations to be generated by altering some shape variables or *parameters* [12].

Numerous techniques for interactive design have been proposed. These include implicit surfaces [8], the B-rep based on polygonal meshes [16] and the very popular spline based modelling schemes such as Bezier and Non-Uniform Rational B-splines (NURBS) [13] found in most 3D modelling software packages.

As far as interactive design using spline based techniques are concerned, they often require large arrays of control points which are difficult and time consuming to manipulate. This is especially true when the designer

Send offprint requests to:

is working in interactive environments. There has been a lot of effort by many researchers to make spline based techniques more intuitive and user friendly. Some of the notable techniques proposed include the technique of introducing constraints on the control points [15, 28], the use of deformations [20] and the use of Dynamic NURBS or D-NURBS [23].

Among other techniques that have been proposed for interactive computer graphics applications include the use of both explicit and implicit integration techniques [19, 2, 1], modal analysis [22, 18] and boundary integral formulations based on Green's function solutions of Navier Stokes equation [17, 10]. Both explicit and implicit integration techniques have been utilised for interactive modelling of soft objects and biomaterials and haptic systems whilst modal analysis for electrodynamics has been reported to be effective for use in interactive force feedback systems. Furthermore, recently the use of boundary integral techniques based on well understood foundations in potential theory have been utilised to for interactive modelling of particular geometries with constraint configurations.

In contrast to the above techniques for interactive design, the method used here uses solutions to elliptic partial differential equations. Dubbed the PDE Method, it works by solving a boundary-value problem with boundary conditions imposed around the edge of the surface patch. PDE surfaces have recently emerged as a powerful modelling technique [11]. For the majority of work carried out on the PDE method, the biharmonic equation, (i.e. the fourth order elliptic PDE), is used

$$\left(\frac{\partial^2}{\partial u^2} + a^2 \frac{\partial^2}{\partial v^2} \right)^2 \mathbf{X}(u, v) = 0 \quad (1)$$

For a review of prior work on the PDE method, the interested reader is referred to [11], where several modelling advantages are also discussed. These include the 'smooth' nature of PDE surfaces, and the fact that they can be parameterised in an efficient manner. However, the point is raised that, current PDE-based techniques have restrained topological structure of modelled objects and lack interactive techniques for direct shape manipulation. To address the problem, Du and Qin [11], developed a toolkit of interactive techniques which rely on a range of numeric methods.

Since analytical expressions of a surface are much more efficient to compute than numerical forms, analytical solutions are always preferred, especially in interactive computer graphics where computational performance is of paramount significance. The order of the PDE determines the number of derivatives of the unknown function $\mathbf{X}(u, v)$ that must be specified in the boundary conditions and hence the degree of continuity that can be obtained between adjoining surfaces. For example, if curvature continuity is required, then solutions to the sixth-order equations need, in general, to be sought.

The work in this paper incorporates the *Spectral Approximations* [6] method of accommodating general boundary conditions into the sixth order PDE solution. Looking at higher order PDEs gives an ability to produce

a wider range of surfaces, while maintaining the versatility of the PDE method. By focusing on surfaces which are periodic, so that, topologically, they are like a ‘closed band’, the PDE has a closed-form analytic solution. This makes the surface computationally very efficient and therefore suitable for use in interactive environments.

A compact user interface is presented which makes use of direct manipulation and other techniques for 3D interaction, to show how the sixth order PDE surface can be manipulated in an interactive environment. This entails extending existing techniques [26] to handle the extra degrees of freedom. It has always been a difficult job to design 3D free-form shapes with basically 2D tools [9], which means that the task of designing a 3D user interface is not an easy one. However it is crucial to have a good user interface to realise the potential of the PDE method.

This paper is ordered in the following way. Section 2 deals with the mathematics of the PDE method, firstly formulating, and then solving the ‘tri-harmonic’ PDE surface. Section 3 discusses the effect of the boundary conditions upon the PDE surface shape, by illustrating similarities to properties of the Hermite curves. In Section 4 it is shown how PDE surfaces can be manipulated using the technique discussed by giving some practical examples. Finally Section 5 focuses on presenting a suitable user interface for the proposed technique.

2 Design formulation using the sixth order PDE

Mathematically, the surface is given by a function $\mathbf{X}(u, v)$ such that,

$$\mathbf{X}(u, v) = (x(u, v), y(u, v), z(u, v)),$$

where u and v are parameters for a finite two-dimensional region Ω , which map onto a point in physical space. For the work described here, a surface patch is obtained by solving the partial differential equation,

$$\left(\frac{\partial^2}{\partial u^2} + a^2 \frac{\partial^2}{\partial v^2} \right)^3 \mathbf{X}(u, v) = 0, \quad (2)$$

for each of the Cartesian coordinates x , y and z . The ‘smoothing parameter’ a controls the relative smoothing of the dependent variables between the u and v directions. Altering a changes the length scale over which the boundary conditions influence the interior of the surface. For more details see [25].

Taking the parameter space Ω to be the region $\{u, v : 0 \leq u \leq 1; 0 \leq v \leq 2\pi\}$ and a solution that is periodic in v , the solution can be obtained analytically. Equation (2) is elliptic and is solved subject to boundary conditions of the form,

$$\mathbf{X}(0, v) = \mathbf{f}_0(v), \quad (3)$$

$$\mathbf{X}(1, v) = \mathbf{f}_1(v), \quad (4)$$

$$\mathbf{X}_u(0, v) = \mathbf{g}_0(v), \quad (5)$$

$$\mathbf{X}_u(1, v) = \mathbf{g}_1(v), \quad (6)$$

$$\mathbf{X}_{uu}(0, v) = \mathbf{h}_0(v), \quad (7)$$

$$\mathbf{X}_{uu}(1, v) = \mathbf{h}_1(v), \quad (8)$$

where subscript u denotes a partial derivative with respect to u . For the following solution, all the given boundary conditions are required to be continuous, and closed in the sense that $\mathbf{f}_0(0) = \mathbf{f}_0(2\pi)$.

2.1 Mathematical Solution

Applying the method of separation of variables to the partial differential Equation (2), gives the general solution,

$$\mathbf{X}(u, v) = \mathbf{A}_0(u) + \sum_{n=1}^N [\mathbf{A}_n(u) \cos(nv) + \mathbf{B}_n(u) \sin(nv)], \quad (9)$$

where the ‘coefficient’ functions $\mathbf{A}_n(u)$ and $\mathbf{B}_n(u)$ are of the form,

$$\mathbf{A}_0(u) = \mathbf{a}_{00} + \mathbf{a}_{01}u + \mathbf{a}_{02}u^2 + \mathbf{a}_{03}u^3 + \mathbf{a}_{04}u^4 + \mathbf{a}_{05}u^5, \quad (10)$$

$$\begin{aligned} \mathbf{A}_n(u) = & \mathbf{a}_{n0}e^{anu} + \mathbf{a}_{n1}e^{-anu} + \mathbf{a}_{n2}ue^{anu} + \\ & + \mathbf{a}_{n3}ue^{-anu} + \mathbf{a}_{n4}u^2e^{anu} + \mathbf{a}_{n5}u^2e^{-anu}, \end{aligned} \quad (11)$$

$$\begin{aligned} \mathbf{B}_n(u) = & \mathbf{b}_{n0}e^{anu} + \mathbf{b}_{n1}e^{-anu} + \mathbf{b}_{n2}ue^{anu} + \\ & + \mathbf{b}_{n3}ue^{-anu} + \mathbf{b}_{n4}u^2e^{anu} + \mathbf{b}_{n5}u^2e^{-anu}. \end{aligned} \quad (12)$$

Using Fourier analysis, the boundary conditions $\mathbf{f}_0(v)$, $\mathbf{f}_1(v)$, $\mathbf{g}_0(v)$, $\mathbf{g}_1(v)$, $\mathbf{h}_0(v)$, $\mathbf{h}_1(v)$ can be written in the form,

$$\mathbf{f}_0(v) = \mathbf{c}_{00}(u) + \sum_{n=1}^N [\mathbf{c}_{n0}(u) \cos(nv) + \mathbf{d}_{n0}(u) \sin(nv)], \quad (13)$$

$$\mathbf{f}_1(v) = \mathbf{c}_{01}(u) + \sum_{n=1}^N [\mathbf{c}_{n1}(u) \cos(nv) + \mathbf{d}_{n1}(u) \sin(nv)], \quad (14)$$

$$\mathbf{g}_0(v) = \mathbf{c}_{02}(u) + \sum_{n=1}^N [\mathbf{c}_{n2}(u) \cos(nv) + \mathbf{d}_{n2}(u) \sin(nv)], \quad (15)$$

$$\mathbf{g}_1(v) = \mathbf{c}_{03}(u) + \sum_{n=1}^N [\mathbf{c}_{n3}(u) \cos(nv) + \mathbf{d}_{n3}(u) \sin(nv)], \quad (16)$$

$$\mathbf{h}_0(v) = \mathbf{c}_{04}(u) + \sum_{n=1}^N [\mathbf{c}_{n4}(u) \cos(nv) + \mathbf{d}_{n4}(u) \sin(nv)], \quad (17)$$

$$\mathbf{h}_1(v) = \mathbf{c}_{05}(u) + \sum_{n=1}^N [\mathbf{c}_{n5}(u) \cos(nv) + \mathbf{d}_{n5}(u) \sin(nv)]. \quad (18)$$

The vector constants of the general solution \mathbf{a}_{ni} and \mathbf{b}_{ni} ($i = 0 \dots 5$), are obtained from the vector constants \mathbf{c}_{ni} and \mathbf{d}_{ni} of the Fourier terms in the boundary conditions. Taking each dimension x , y and z separately, and each Fourier modes $n = \{0, \dots, N\}$ in turn,

$$\begin{pmatrix} \mathbf{a}_{n0} \\ \vdots \\ \mathbf{a}_{n5} \end{pmatrix} = \mathbf{M} \begin{pmatrix} \mathbf{c}_{n0} \\ \vdots \\ \mathbf{c}_{n5} \end{pmatrix} \quad (19)$$

and

$$\begin{pmatrix} \mathbf{b}_{n0} \\ \vdots \\ \mathbf{b}_{n5} \end{pmatrix} = \mathbf{M} \begin{pmatrix} \mathbf{d}_{n0} \\ \vdots \\ \mathbf{d}_{n5} \end{pmatrix}, \quad (20)$$

where $\mathbf{M}(a, n)$ is a matrix, equivalent to a direct comparison of coefficients. However, in many cases, an infinite Fourier series ($N = \infty$) may be required to represent the boundary conditions exactly. So, in order to circumvent the problem of having to calculate an infinite series, the following observation is used. If the coefficient functions $\mathbf{A}_n(u)$, $\mathbf{B}_n(u)$ are written in the form

$$(\mathbf{k}_0 + \mathbf{k}_2 u + \mathbf{k}_4 u^2)e^{an(u-1)} + (\mathbf{k}_1 + \mathbf{k}_3 u + \mathbf{k}_5 u^2)e^{-an(u-1)}, \quad (21)$$

where $\mathbf{k}_0, \dots, \mathbf{k}_5$ are vector-valued constants, then it can be seen that the first part decays exponentially with u away from the boundary $u = 1$ and the second part decays exponentially with u away from the boundary $u = 0$. For larger values of an , the decay is more rapid. Therefore, for large enough n , the Fourier modes make negligible contributions to the interior of the patch.

Hence it is reasonable to truncate the Fourier series at some finite N , (typically $N = 5$ is adequate), and represent the contribution of the high frequency modes to the surface with a remainder function $\mathbf{R}(u, v)$. In this paper $\mathbf{R}(u, v)$ is chosen to be of the form

$$\mathbf{R}(u, v) = \mathbf{r}_0 e^{\omega u} + \mathbf{r}_1 e^{-\omega u} + \mathbf{r}_2 u e^{\omega u} + \mathbf{r}_3 u e^{-\omega u} + \mathbf{r}_4 u^2 e^{\omega u} + \mathbf{r}_5 u^2 e^{-\omega u}. \quad (22)$$

The coefficient functions $\mathbf{r}_0(v), \dots, \mathbf{r}_5(v)$ are calculated from the difference between the real boundary conditions (3-8) and approximate boundary conditions (13-18).

$$\mathbf{R}(0, v) = \mathbf{f}_0(v) - \mathbf{X}(0, v), \quad (23)$$

$$\mathbf{R}(1, v) = \mathbf{f}_1(v) - \mathbf{X}(1, v), \quad (24)$$

$$\mathbf{R}_u(0, v) = \mathbf{g}_0(v) - \mathbf{X}_u(0, v), \quad (25)$$

$$\mathbf{R}_u(1, v) = \mathbf{g}_1(v) - \mathbf{X}_u(1, v), \quad (26)$$

$$\mathbf{R}_{uu}(0, v) = \mathbf{h}_0(v) - \mathbf{X}_{uu}(0, v), \quad (27)$$

$$\mathbf{R}_{uu}(1, v) = \mathbf{h}_1(v) - \mathbf{X}_{uu}(1, v). \quad (28)$$

An interesting point is that this choice of $\mathbf{R}(u, v)$, allows the coefficient functions to be calculated using the same matrix $\mathbf{M}(a, n)$ as that used previously

(19-20), which makes the whole procedure very computationally efficient. Thus, a surface patch $\tilde{\mathbf{X}}(u, v)$ is represented by

$$\tilde{\mathbf{X}}(u, v) = \mathbf{X}(u, v) + \mathbf{R}(u, v).$$

The choice of N will affect how well $\tilde{\mathbf{X}}(u, v)$ approximates the solutions of the original partial differential Equation (2). However, it should be noted that $\tilde{\mathbf{X}}(u, v)$ shares many properties of the exact solution, including the fact that they both satisfy the same set of boundary conditions.

The main aim here is to obtain solutions for the 6th order PDE surfaces with general boundary conditions rapidly enough to be able carry out interactive design in real-time. The above solution method can be viewed as a spectral method for the solutions of PDEs where the expansion functions are based upon the Fourier series. This solution scheme can be applied in the case of general complex periodic boundary conditions. For the majority of practical designs the method therefore is suitable. In the case of non-periodic boundary conditions, an analytic approximation for the PDE can still be found [4], although it has not been implemented in this work.

2.2 Parameters of the PDE surface

The parameter a in Equation (2) determines the rate at which the solution ‘forgets’ about the boundary condition as one moves away from a boundary into the surface interior. In choosing $\mathbf{R}(u, v)$ to be of the form of (22), we are solving (1) with $a = a(n)$. More precisely

$$a = \begin{cases} a(\text{constant}) & n \leq N, \\ \omega/n & n > N, \end{cases}$$

which shows that the parameter ω determines the rate at which the solution ‘forgets’ about the higher order Fourier modes. (i.e. the features of the boundary conditions which are expressed by the lower Fourier modes, such as corners). Since the $(N + 1)$ mode is the dominant mode of those ignored, choosing $\omega = a(N + 1)$ gives a decay rate close to the actual decay rate of the exact solution. However there is nothing preferred about this choice of ω , and so its value can be left for the user to determine in the same manner as the smoothing parameter a . With ω controlling the small scale features, and a controlling the long-length scale features. Also the choice of N affects the shape of the generated surface. However the effect is not useful as a design tool, and N is usually chosen as a trade off between speed of calculation and the level of approximation to the exact solution. However, altering these parameters is a secondary effect. The main control of a surface is gained through manipulating the boundary conditions.

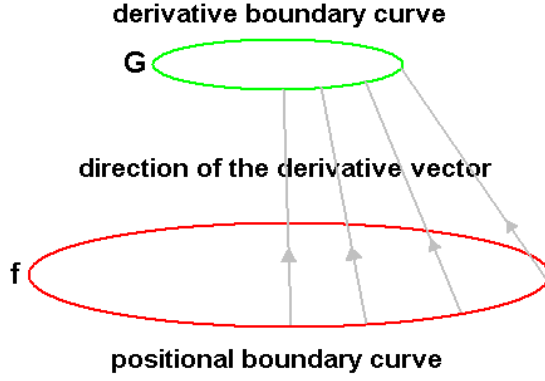


Fig. 1 Derivative vector field.

3 Specifying boundary conditions

Since the PDE method treats surface design as a boundary value problem, the boundary conditions play a vital role in determining the shape of the surface. However, it is clear that the designer cannot be expected to be familiar with the mathematics of elliptic PDEs. So Ugail *et al* [25] developed a methodology to specify the periodic boundary conditions for the ‘biharmonic’ PDE (1) by means of four curves in 3-space. The ‘triharmonic’ PDE (2) requires extra curvature boundary conditions, but the existing approach can be used to specify the positional and first-derivative boundary conditions (3-6). In particular two curves, $\mathbf{f}_0(v)$ and $\mathbf{f}_1(v)$, describe the position of the boundary, i.e. how $\mathbf{X}(u, v)$ varies at the $u = 0$ and $u = 1$ isolines. A vector field to specify the derivatives on the boundary, is given by the difference between points on the ‘positional curves’ and corresponding points on the ‘derivative curves’, as shown in Figure 1.

Thus the ‘derivative boundary conditions’ (5) and (6) are specified using both the magnitude and direction of this vector field. Labelling the derivative curves $\mathbf{G}_0(v)$ and $\mathbf{G}_1(v)$, the formulation of the derivatives can be written as in Equation (29-30).

$$\mathbf{X}_u(0, v) = [\mathbf{f}_0(v) - \mathbf{G}_0(v)]k_1 \quad (29)$$

$$\mathbf{X}_u(1, v) = [\mathbf{G}_1(v) - \mathbf{f}_1(v)]k_1 \quad (30)$$

Note that the tangent vectors at the $u = 1$ end of the surface are calculated in the reverse direction, and the value of the scalar k_1 is chosen for convenience of user interaction.

This method of specifying the derivative boundary conditions is just one of several possible choices. Likewise there are many different ways of specifying the curvature boundary conditions. In this paper, the second derivatives are specified as the difference between the ‘positional curves’



Fig. 2 Family of cubic curves with varying direction of the tangent vector at the left endpoint. The green lines represent tangent vectors at the ends of the red Hermite curves.

and ‘second derivative curves’ ($\mathbf{H}_0(v)$, $\mathbf{H}_1(v)$), in a similar manner to the first derivative.

$$\mathbf{X}_{uu}(0, v) = [\mathbf{f}_0(v) - \mathbf{H}_0(v)]k_2 \quad (31)$$

$$\mathbf{X}_{uu}(1, v) = [\mathbf{f}_1(v) - \mathbf{H}_1(v)]k_2 \quad (32)$$

With appropriately chosen values for the scalar constants k_1 and k_2 , the second derivative has a similar intuitive feel to that of the first derivative, and so it seems reasonable to provide a similar mechanism for controlling them both.

3.1 Surface control given by boundary conditions

To give an understanding of how the boundary conditions affect the geometry of the surface, consider the following simplification. If the boundary conditions (3-8) are simplified to vector constants, then the only term from the general solution (9) that contributes to this particular solution is the \mathbf{A}_0 term (10). For the biharmonic equation, this can be expressed as a cubic Hermite polynomial curve in the u parameter, defined by two end points and two endpoint tangent vectors. Properties of the Hermite curves are briefly examined by following the approach of [14], and then these observations are used to gain insight into how the boundary conditions affect the PDE surface.

Figure 2 shows a series of Hermite curves, with different directions of tangent vector. In an interactive graphics system, the endpoints and tangent vectors of a curve can be manipulated interactively. Additionally, the length of the tangent vectors can be altered to increase or decrease their effect upon the curve, as can be seen in Figure 3. The length of the tangent vector is multiplied by constant k_1 (29-30) in order to make the length of the graphical tangent vectors relate in an intuitive way to the dimensions of the curve.

Similarly, the triharmonic equation is reduced to a quintic polynomial curve when the boundary conditions are taken as vector constants. The additional curvature constraints at the end points, give greater control of the curve. Figure 4a is identical to Figure 2a, but with the extra vectors representing the curvature at the endpoints. Here the curvature vectors are



Fig. 3 Family of cubic curves with varying magnitude of the tangent vector at the left endpoint.

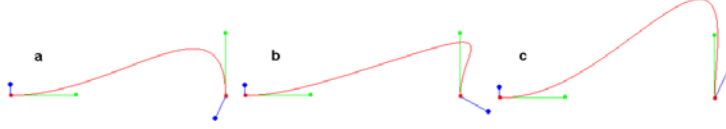


Fig. 4 Family of quintic curves with varying direction of the second-derivative vector (shown as blue lines) at the right endpoint.

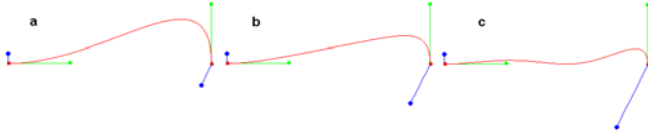


Fig. 5 Family of quintic curves with varying magnitude of the second-derivative vector at the right endpoint.

determined explicitly such that the quintic curve is reduced to its special case of a cubic curve. Figure 4 shows the effect produced upon the quintic curve, by altering the direction of the curvature vector.

Figure 5 demonstrates the effect of adjusting the magnitude of the curvature vector.

The reason for this detailed discussion is that there exist an interesting connection between Hermite surfaces and PDE surfaces in that in the limit $a \rightarrow 0$ the PDE surface turns into a Hermite surface such that curves of constant v are Hermite curves. However, due to the smoothing effect of the PDE ($a \neq 0$) the curves of constant v are not Hermite, but considering them in these terms is useful for the intuitive control of the PDE surface.

The Hermite curve can be seen as being defined by six control points: two end points, two points that control the ends of the tangent vectors and two points for the second derivatives. Thus, a PDE surface can be seen as being controlled by six continuous space curves. These are the curves $\mathbf{f}_0(v)$, $\mathbf{f}_1(v)$, $\mathbf{G}_0(v)$, $\mathbf{G}_1(v)$, $\mathbf{H}_0(v)$, $\mathbf{H}_1(v)$, used to define the boundary conditions. This concept is demonstrated in more detail by looking at an example where a cylindrical PDE surface is deformed.

4 PDE surface manipulation

The cylindrical PDE surface in Figure 6a is created by choosing six circles (defined here as closed B-Spline curves) as the boundary curves, and aligning them along an axis. The green and blue lines represent the tangent and second-derivative vectors at certain representative points around the boundaries. To illustrate the various effects of the different boundary conditions on the shape of a surface, one of the cross-sectional curves is deformed, by altering each different node in turn.

Note that in order to keep the tangent and the second derivative vectors fixed, the derivative and second-derivative curves are also deformed. Figure 6c shows the effect of modifying the tangent vector of the initial surface, while Figure 6d shows a similar manipulation carried out upon the second-derivative vector. The shape of the deformation is similar in both cases, except that the tangent vector has an effect closer to the surface edge, than the second-derivative vector. The region affected by each type of manipulation is determined by the Hermite blending functions, shown in Figure 7.

The influence of the tangent function \mathbf{G} is similar to that of the second-derivative function \mathbf{H} , which is the reason why they have similar intuitive control over the surface shape. It is possible to use different blending functions by changing Equations (29-32) which determine how the derivative boundary conditions are controlled from the boundary curves. One example would be to use Bernstein polynomials [14] used by Bézier curves which have the added *convex hull* property. However, this would require a different user interface ‘metaphor’ for controlling the curvature constraints.

4.1 Examples

The surface shown in Figure 6 is very simple in comparison to the range of geometries that it is possible to create with a sixth order PDE patch. A more general example, is the ‘hipflask or shampoo bottle’ in Figure 8a, which was created with a single PDE patch.

Figure 8b shows the boundary curves that are used to generate the hipflask shape. To facilitate appreciation of the effect of these lines, Figure 8c shows how the same boundary conditions are represented in the vector form as described earlier.

In earlier work using the biharmonic PDE, two patches were joined together with curvature continuity to model fluid membranes and vesicles. An optimisation for minimum surface energy was performed directly on the constants in Equations (13-18), to obtain several different vesicle shapes [5]. By interactively specifying the starting shape, it would be possible to find more of these shapes. Figure 9 shows two interactively created, vesicle shapes, using only a few parameters. These initial shapes are therefore suitable to be used in a numerical optimisation scheme, to refine their shape based on a functional criteria.

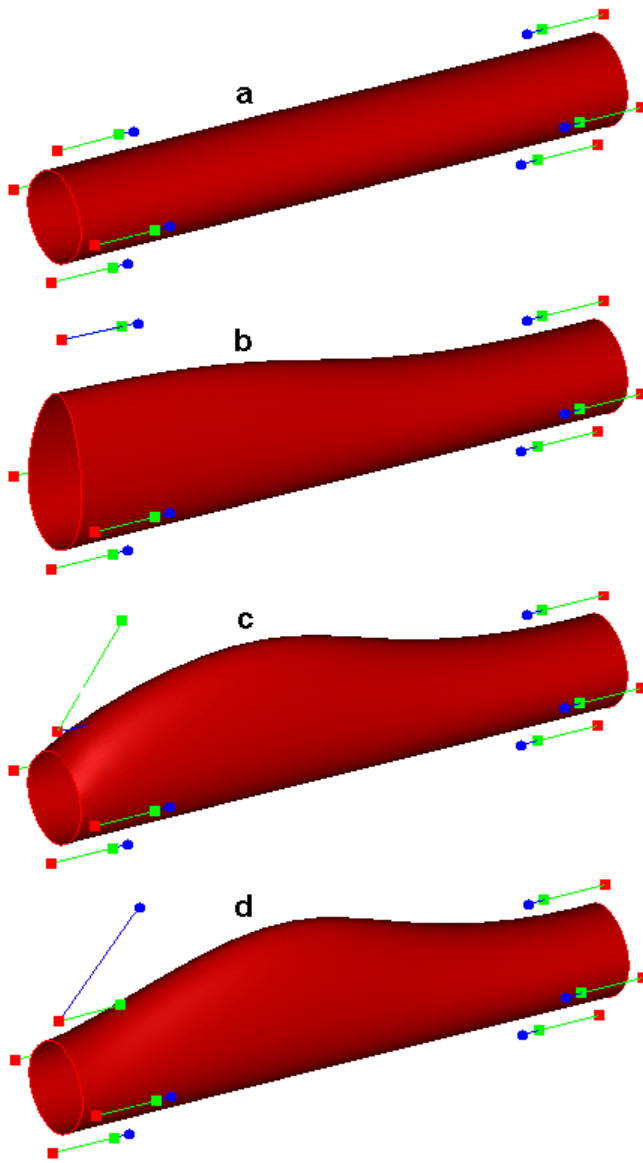


Fig. 6 The effect of altering a positional node(b), derivative vector(c), or a second-derivative vector(d) on the initial PDE surface(a).

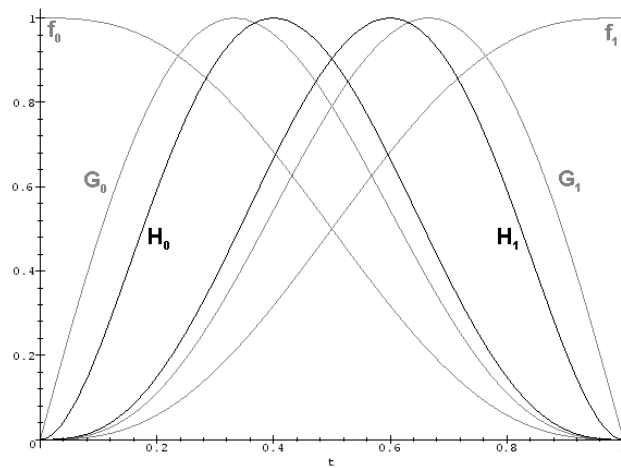


Fig. 7 Blending functions for sixth order Hermite curve, labelled by the space curve which they weight.

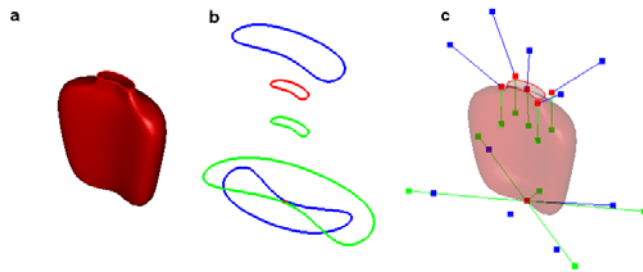


Fig. 8 Sixth Order PDE surface patch modelling a Hipflask, or Shampoo Bottle.

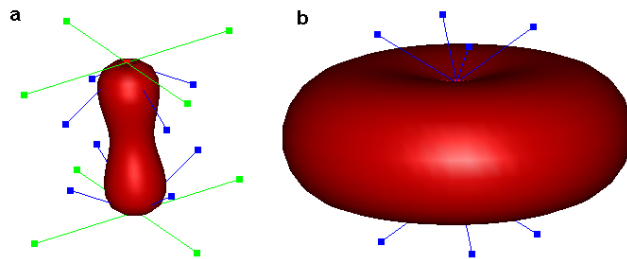


Fig. 9 Sixth Order PDE surface patch modelling Vesicles.

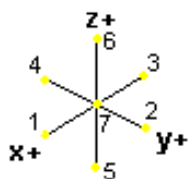


Fig. 10 Local coordinate system with control points that can be manipulated by a cursor.

It is noteworthy that the shape is parameterised by the way in which the boundary curves are parameterised. This could in principle be done through the Fourier representation of the curves as done in [5]. However, interactive design is better carried out by imposing a specified geometry on each curve which can be parameterised in terms of the associated control points. Each of the control points used in the interactive design, are visual representations of the design parameters. Adjustments of the design parameters allow the user to select from a whole range of possible designs once an initial generic parameterisation has been specified. This technique is very general since the remainder term (22) guarantees the boundary conditions are exactly satisfied for any closed parametric curve representation. By carefully choosing each curve type to contain a minimum number of shape parameters, while maintaining sufficient generality, a surface may be parameterised efficiently. Furthermore, choosing curves which have parameters that are geometrically intuitive, lends itself to efficient interactive design.

5 User interface for interactive design

In order to realise the potential of PDE based modelling scheme as a practical design tool, it is paramount to have an intuitive user interface so that the mathematical details are hidden from view.

In the PDE based modelling scheme a surface is controlled mainly by its curves, and so a method of controlling space curves can be applied directly to a surface. In order to provide a consistent user interface for a wide range of curve types, a curve is defined as a container of 3D points with a map P such that $P : \mathbb{R} \rightarrow \mathbb{R}^3$.

Direct manipulation of the control points of a 3D curve is difficult using a 2D input device such as a mouse. To overcome this problem the interactive technique proposed by Bier [3] is taken. This technique allows the specification of all the nine transformation parameters associated with translation, rotation and scaling in 3D. The parameters are specified with the aid of a projected Cartesian coordinate frame called a *jack*. A jack consists of seven virtual control points, one in the centre and six at the ends of the axes as shown in Figure 10.

The user can pick at a virtual control point and drag it on the surface of the screen. The 2D mouse movement is converted to a 3D object transformation by matching the movement with the 2D projection of the jack on the screen. For example a down and left movement on the jack in Figure 10 is mapped to the x -axis. A translation is performed by dragging the centre of a jack, along one of its projected axes. Scaling is specified by dragging the end of an axis towards or from the centre of the coordinate system. Rotation is specified by dragging a control point at the end of an axis parallel to one of the two other axes, using the centre of the jack as the anchor of the rotation. A restriction of this system is that transformations can only be specified in one direction at a time. More complex transformations, such as a uniform scaling or a translation in a plane, have to be specified as a sequence of the basic transformations. However, there is an advantage here, since the different x , y and z components are calculated independently of each other, if a manipulation does not affect a particular direction, it does not need to be recalculated.

The proposed method of controlling the space curves, is to use a number of jacks as the control points, in a similar manner to Emmerik's GeoNode modelling system [12]. The difference being, that jacks are only drawn at *selected* control points. Any transformation carried out effects the entire current-selection. This enables manipulation of portions of a curve or even several curves in one step.

5.1 Specifying new curves and surfaces

The technique described above is used for manipulating curves and surfaces. However the task of specifying new points for a curve, requires a different mechanism. The approach taken by Ugail [26] was to first specify a curve in the $x - y$ plane, and then transform the z coordinate afterwards. To generalise that approach, the front faces of the viewing volume cube are used as the initial planes, which any mouse click is projected onto. Once an initial position is specified, a jack is created at that point, and then a subsequent drag will specify the third coordinate. This allows specification of new points for any 3D location, in one movement.

A mechanism for specifying precise locations is also desirable. To achieve this, a 'data entry box' provide feedback on the numerical position of the control point being manipulated (or the average position of all the selected nodes in the case of multiple selection). Making use of the 'Equal Opportunity' design heuristic [21], the output of the numerical position, can also be used as input. This allows a user to type in the exact location of any control point, if precision is required.

5.2 Interactive shape manipulation

When shape manipulation is carried out interactively, for the interest of clarity, the tangent and second-derivative curves are usually hidden from

view. Also, at the initial stage of the design, they are pre-determined to be of the same type as the positional curve, including the same number of nodes. This allows the tangent and second-derivative vectors to be drawn as short lines, as shown in Figure 6. This is a restriction aimed to help the novice user, by simplifying the surface to work in a predictable way. Furthermore, this technique enables the user to maintain a consistent parameterisation between the boundary curves.

Also at the initial stage of a design process the user is given the freedom of starting out a design with a fourth order PDE solution. Should he/she require higher order shape manipulation the fourth order PDE patch can be automatically converted to a sixth order surface patch where the initial values for the curvature conditions can be obtained from the curvature at the boundaries of the fourth order PDE.

In the course of designing a surface, it is quite common that the control points become obscured by the surface itself. The surface may be drawn transparent (either accumulation buffer, or polygon stipple) or in wire frame mode, to help with this [27]. Since the model of the surface is 3D it is also necessary to be able to change the position and orientation of the viewpoint. A useful technique of mapping 2D mouse movements into 3D rotations is to make use of Quaternions, [14]. In addition to this, a reference framing cube surrounding the working space is used to provide location feedback. This viewing volume cube, can itself be manipulated, using jacks again, to alter the view.

6 Conclusions

The discussion on this paper has been focused on extending the PDE method for interactive design of complex shapes. In contrast to previous work the governing partial differential equation has been generalised to sixth order in order to introduce more flexibility. An existing analytic method for solving the fourth order PDE has been extended to cater for higher order PDEs. This allows the PDE to be solved efficiently, even in the case of general boundary conditions, making the technique fast enough for real time interactive design.

In the case of PDE surfaces the shape manipulation is essentially carried out using the boundary curves that represent the shape. By seeking a connection between Hermite curves and the cross-sectional curves which form PDE surfaces, a technique has been proposed for efficient manipulation of surfaces. Furthermore by making use of direct manipulation and other techniques for 3D interaction an intuitive user interface is presented for effective manipulation of PDE surfaces using its boundary curves.

The essence of this paper is to demonstrate how the sixth order PDE can be utilised for interactive design, in particular we have shown how the 6th order boundary conditions can be chosen for interactive design. For the purpose of demonstration we have chosen relatively simple examples.

Previous work mainly used 4th order PDEs where simply the function and derivative boundary conditions were specified. Even so, shapes of considerable complexity have been produced e.g. propellor blades [7], aircrafts [6] and biological membranes [5]. Essentially such topologically complex objects were created by means of using a small number of smoothly connected PDE surface patches. The use of 6th PDE, therefore, not only allows the capability of generating surface patches the 4th order PDEs can generate but also even more complex surface patches due to the two added curvature boundary conditions.

One could also argue that the examples we have highlighted in this paper, in particular the ‘tube like’ example shown in Figure 6 could be generated using other simpler methods such as lofting. While this is true the smoothness and continuity requirements of shapes generated using PDEs are much more well suited for many design applications. In fact, the surfaces generated using analytic solutions to PDEs as described in this paper are infinitely differentiable even when discontinuities are present at the boundaries.

One of the main advantages of using PDE surfaces is that the shapes represented using PDE surfaces have an efficient parameterisation associated with them. An interesting new direction of study will be the investigations into the development of an efficient parametric model for higher order PDEs by making use of the boundary curves so that a designer sitting in front of a workstation can quickly change the designs by interacting with them in real time.

Acknowledgements The authors wish to acknowledge the financial support for S. Kubiesa from EPSRC of UK through which this work was completed.

References

1. O. Astley and V. Hayward 1998. Multirate haptic simulation achieved by coupling finite element meshes through Norton equivalents. *In Proc. IEEE Int. Conf. on Robotics and Automation*, 1998. 989–994.
2. D. Baraff and A. Witkin. Large steps in cloth simulation. *In SIGGRAPH 98. Conference Proceedings*, 1998. 43–54.
3. E. A. Bier. Skitters and jacks: Interactive 3D positioning tools. In *Workshop on Interactive 3D Graphics*, University of North Carolina, Chapel Hill, 1986 183–196.
4. M. I. G. Bloor and M. J. Wilson. An analytic pseudo-spectral method to generate regular 4-sided PDE surface patch. *Computer-Aided Geometric Design*, To Appear, 2004.
5. M. I. G. Bloor and M. J. Wilson. Method for efficient shape parametrization of fluid membranes and vesicles. *Physical Review E, Part B*, **61**(4), 2000 4218–4229.
6. M. I. G. Bloor and M. J. Wilson. Spectral approximations to PDE surfaces. *Computer-Aided Design*, **28**(2), 1996 145–152.

7. C. W. Dekanski, M. I. G. Bloor, and M. J. Wilson. The generation of propeller blade geometries using the PDE method. *Journal of Ship Research*, **39**(2), 1995 108–116.
8. M. Desbrun, N. Tsingos and M. P. Cani-Gascuel. Adaptive Sampling of Implicit Surfaces for Interactive Modelling and Animation. *Computer Graphics Forum*, **15**(5), 1994 319–327.
9. C. G. C. van Dijk and A. A. C. Mayer. Sketch input for conceptual surface design. *Computers in Industry*, **34**, 1997 125–137.
10. D. G. Duffy. *Green's Functions with Applications*, Chapman and Hall, Boca Raton, Florida 2001.
11. H. Du and H. Qin. Dynamic PDE surfaces with flexible and general geometric constraints. In B. A. Barsky, Y. Shinagawa, and W. Wang, editors, *Proceedings of the 8th Pacific Graphics Conference on Computer Graphics and Application*, IEEE, 213–223.
12. M. J. G. M. van Emmerik. *Interactive design of parameterized 3D models by direct manipulation*. Delft University Press, November 1990.
13. G. Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press, 1998.
14. J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer graphics: Principles and practice*. Addison-wesley, 1997.
15. B. Fowler. Geometric Manipulation of Tensor Product Surfaces, In *Computer Graphics (1992 Symposium on Interactive 3D Graphics)*, D. Zeltzer (ed.), **25**, 1992 101–108.
16. J. Hoschek and D. Lasser. *Computer Aided Geometric Design*. A K Peters, 1993.
17. D. James and D. K. Pai. ArtDefo, accurate real time deformable objects, In *Computer Graphics, ACM SIGGRAPH 99 Conference Proceedings*, 1999 65–72.
18. J. F. O'Brien and J. K. Hodgins. Graphical modeling and animation of brittle fracture. In *Proceedings of SIGGRAPH 99*, 1999 137–46.
19. G. Picinbono, H. Delingette, and N. Ayache. Non-Linear Anisotropic Elasticity for Real-Time Surgery Simulation. *Graphical Models*, **65**(5), 2003, 305–321.
20. A. Rappoport, Y. Helor and M. Werman. Interactive Design of Smooth Objects with Probabilistic Point Constraints *ACM Transactions on Graphics*, **13**, 1994 156–176.
21. C. Runciman and H. Thimbleby. Equal opportunity interactive systems. *International Journal of Man-Machine Studies*, **25**(4), 1986 439–451.
22. J. Stam. Stochastic dynamics: simulating the effects of turbulence on flexible structures, *Computer Graphics Forum, EUROGRAPHICS'97 Proceedings*, **16** 3, 1997 159–164.
23. D. Terzopoulos and H. Qin. Dynamic NURBS with geometric constraints for interactive sculpting. *ACM Transactions on Graphics*, **13**, 1994 103–136.
24. H. Thimbleby. *User interface design*. ACM Press, 1990.
25. H. Ugail, M. I. G. Bloor, and M. J. Wilson. Techniques for interactive design using the PDE method. *ACM Transactions on Graphics*, **18**(2), 1999 195–212.
26. H. Ugail, M. I. G. Bloor, and M. J. Wilson. Manipulation of PDE surfaces using an interactively defined parameterisation. *Computers and Graphics*, **4**(23), 1999 525–534.
27. M. Woo, J. Neider, T. Davis, and D. Shreiner. *OpenGL Programming Guide*. Addison-wesley, 1999.

- 28. H. Xie and H. Qin, Automatic Knot Determination of NURBS for Interactive Geometric Design, *In Proceedings of International Conference on Shape Modelling and Applications (SMI 2001)*, 2001.
- 29. H. Yamashina, K. Fukushima, and A Saijo. CAD for free-form surfaces. *Computer Integrated Manufacturing Systems*, **9**(1),1996 9–18.